

Anteproyecto aplicación gestión de videojuegos

David Bargalló Ortiz

Desarrollo de Aplicaciones Multiplataforma

07/04/2025

Índice

- [Anteproyecto aplicación gestión de videojuegos](#)
 - [Índice](#)
 - [1. Definición del problema](#)
 - [2. Planteamiento del proyecto](#)
 - [¿Qué necesidades tiene el usuario?](#)
 - [Solución propuesta](#)
 - [Diseño de la aplicación](#)
 - [Fase de implantación](#)
 - [Pruebas](#)
 - [3. Herramientas que usaremos](#)
 - [4. Objetivos](#)
 - [5. Bibliografía](#)
-

1. Definición del problema

El objetivo de este proyecto es dar solución a la desorganización que muchas personas experimentan al gestionar su colección de videojuegos, tanto físicos como digitales, en diferentes plataformas. Se busca centralizar toda la información en una única aplicación, permitiendo una consulta rápida, filtrado avanzado, seguimiento de precios y ubicación física de los juegos.

Además, se resolverán otros problemas como:

- La descentralización de catálogos digitales entre múltiples plataformas.
 - La falta de visibilidad de complementos (DLCs) adquiridos o pendientes.
 - La dificultad para recordar la ubicación física de los juegos.
 - La imposibilidad de tener una visión global de la colección y wishlist.
-

2. Planteamiento del proyecto

¿Qué necesidades tiene el usuario?

- Centralizar su colección de videojuegos en una única plataforma.
- Filtrar la colección por diferentes parámetros (nombre, consola, precio, formato...).
- Consultar DLCs de cada juego y si los posee.
- Almacenar la ubicación física de los juegos físicos.
- Añadir títulos a una wishlist con precios actualizados.

- Exportar su colección y wishlist en formato PDF.
- Ver estadísticas relevantes de sus juegos.

Solución propuesta

La aplicación resolverá estas necesidades mediante:

- Una base de datos que almacena toda la información relevante de los juegos, usuarios y relaciones entre ellos.
- Una API para obtener bibliotecas de juegos por plataforma y para mantener actualizados los precios.
- Interfaz gráfica intuitiva desarrollada con JavaFX y Scene Builder.
- Filtros personalizados para buscar juegos específicos o gestionar la wishlist.
- Exportación de la biblioteca y wishlist a PDF.
- Acceso mediante login para permitir varios usuarios con sus propias bibliotecas (contraseña hasheada).

Diseño de la aplicación

La aplicación contará con estas pantallas:

1. **Pantalla de inicio de sesión:** acceso de usuarios con su nombre de usuario y contraseña.
2. **Pantalla principal:** con:
 - Los últimos juegos añadidos (se decidirá más adelante si serán 3, 5, 10...).
 - Vista previa de la wishlist con precios (listview con el nombre del juego, la consola y precio, la wishlist tendrá una interfaz propia más detallada).
 - Estadísticas destacadas (3-4 estadísticas como preview de la pantalla con todas las estadísticas).
3. **Pantalla de biblioteca:** muestra en fichas todos los juegos del usuario (portada, nombre consola...). Al pulsar en una ficha, se abrirá una nueva ventana con información completa del juego (en la ficha habrá una especie de preview, la ficha entera será una ventana entera con todos los datos).
4. **Pantalla de filtros:** para buscar juegos en la biblioteca o wishlist según los filtros seleccionados.
5. **Pantalla de wishlist:** donde el usuario puede ver los juegos deseados, actualizar precios y ordenarlos por parámetros como consola o precio.
6. **Pantalla de estadísticas:** visualización de estadísticas como número de juegos por plataforma, juegos por consola, juegos por género, etc.
7. **Pantalla de configuración/cambio de contraseña.**
8. **Pantalla de exportación de datos** (Aún no se sabe si una pantalla o un botón en la pantalla de la biblioteca).

Fase de implantación

1. Diseño del modelo Entidad-Relación (ER) de la base de datos.
2. Creación de la base de datos en línea usando Supabase (SQL).
3. Creación de la interfaz de todas las pantallas en Scene Builder.
4. Búsqueda e integración de APIs para obtener catálogos de juegos y precios actualizados.
5. Desarrollo de la interfaz con JavaFX en Visual Studio Code.
6. Programación de la lógica de negocio: conexión con la BBDD, tratamiento de datos, integración con API.
7. Realización de pruebas unitarias, funcionales y de rendimiento.
8. Elaboración de documentación técnica y manual de usuario (durante todo el desarrollo).

Pruebas

- Verificación de todas las funcionalidades (CRUD de juegos, filtros, login, wishlist, estadísticas...).
 - Validación de conexiones con BBDD y APIs.
 - Comprobación de rendimiento con gran cantidad de datos.
 - Pruebas con usuarios reales para recoger feedback (si hay tiempo suficiente y se puede hacer).
-

3. Herramientas que usaremos

(Este apartado está sujeto a cambios durante el desarrollo)

- **Sistema operativo:** Windows
 - **Tipo de aplicación:** Escritorio (JavaFX)
 - **IDE:** Visual Studio Code
 - **Lenguaje de programación:** Java
 - **Gestión de dependencias:** Maven
 - **Gestión de versión:** Git + GitHub
 - **Diseño de interfaz:** JavaFX + Scene Builder
 - **Base de datos:** Supabase (SQL en la nube)
 - **APIs utilizadas:**
 - RAWG o alternativa para obtener bibliotecas de juegos y detalles.
 - PriceCharting o alternativa para precios actualizados.
 - **Pruebas unitarias:** JUnit
 - **Pruebas de API:** Postman
-

4. Objetivos

- Permitir a los usuarios gestionar y consultar su colección de videojuegos.
 - Implementar un sistema de login y gestión por usuarios.
 - Añadir funcionalidades avanzadas como wishlist, seguimiento de precios, estadísticas y exportación.
 - Ofrecer una experiencia visual amigable mediante fichas gráficas y filtros intuitivos.
 - Integrar APIs externas para automatizar la carga de datos y actualización de precios.
 - Permitir la exportación de la colección en formato PDF.
 - Desarrollar pruebas automatizadas para asegurar la calidad del software.
-

5. Bibliografía

- ChatGPT. (n.d.). *ChatGPT*. Recuperado el X de abril de 2025, de <https://chatgpt.com/>
- Gemini. (n.d.). *Gemini*. Recuperado el X de abril de 2025, de <https://gemini.google.com/app>
- Google. (n.d.). *Google*. Recuperado el X de abril de 2025, de <https://www.google.es/>
- Stack Overflow. (n.d.). *Stack Overflow*. Recuperado el X de abril de 2025, de <https://stackoverflow.com>
- YouTube. (n.d.). *YouTube*. Recuperado el X de abril de 2025, de <https://www.youtube.com/>
- RapidAPI. (n.d.). *RapidAPI*. Recuperado el X de abril de 2025, de <https://rapidapi.com/>
- Documentación oficial de Java, JavaFX, Scene Builder, Supabase y Maven.